

Implementación en hardware reconfigurable de un controlador para una cámara CCD

Juan Moisés Arredondo Velázquez

Instituto Tecnológico de Celaya, Av. Tecnológico y Av. García Cubas s/n. Celaya, Gto. México, (461) 6117575

Javier Díaz Carmona

Instituto Tecnológico de Celaya, Av. Tecnológico y Av. García Cubas s/n. Celaya, Gto. México, (461) 6117575

javier.diaz@itcelaya.edu.mx

Alfredo Padilla Medina

Instituto Tecnológico de Celaya, Av. Tecnológico y Av. García Cubas s/n. Celaya, Gto. México, (461) 6117575

Agustín Ramírez Agundis

Instituto Tecnológico de Celaya, Av. Tecnológico y Av. García Cubas s/n. Celaya, Gto. México, (461) 6117575

Noé Vela Aguirre

Instituto Tecnológico de Celaya, Av. Tecnológico y Av. García Cubas s/n. Celaya, Gto. México, (461) 6117575

Resumen

El procesamiento de imágenes comúnmente requiere una gran cantidad de recursos en cuanto a hardware se refiere, además de un enorme conjunto de operaciones, por tanto el procesamiento en paralelo realizado por hardware especializado se vuelve indispensable. En muchas ocasiones el algoritmo a implementar depende en gran parte del sensor de imagen con el que se cuente. En este artículo se presenta un controlador en hardware reconfigurable para un sensor de imagen tipo CCD (Charge Coupled Device), con el cual es posible configurar funciones disponibles. Entre dichas funciones el usuario puede seleccionar una de las tres longitudes de onda o incluso obtener información de los tres mapas de color (RGB).

Palabras Claves: Cámara CCD, Controlador en hardware reconfigurable, Memoria de doble puerto.

1. Introducción

Para muchas aplicaciones de procesamiento de imágenes implementadas en hardware reconfigurable, una de las primeras consideraciones es la selección del sensor de imagen a utilizar y muchas de las ocasiones el sensor no cuenta con el controlador propio para ser implementado en un FPGA. La mayoría de las ocasiones dichos sensores contienen un controlador que puede ser implementado y probado, pero el inconveniente es que dichos controladores solo sirven para demostrar el funcionamiento del sensor que en la mayoría de los casos son muy complejos. Entender la lógica implementada o incluso modificar la descripción del controlador se vuelve una tarea que demanda mucho tiempo, además al no estar documentados estos controladores, se vuelve una desventaja mayor.

El diseño de un controlador de una cámara CCD es una tarea tan complicada como la aplicación lo demande. En [1] se detalla el diseño de un controlador genérico implementado en un FPGA, donde la lógica es controlada mediante una ASM (Algorithmic State Machine) derivada de un código propio para un DSP. De igual manera existen diversos trabajos basados en el desarrollo de controladores genéricos facilitando la implementación en hardware de algoritmos específicos en el procesamiento de imágenes. En [2] se presenta otra alternativa para el diseño de un controlador, en donde el código se diseña en simulink y se implementa en un FPGA con el fin de aprovechar todo el potencial que una cámara CCD puede entregar. Además del desarrollo de controladores, también es importante considerar el efecto de la circuitería interna de una cámara CCD; en [3] se propone el diseño de un chip de bajo consumo de energía para controlar digitalmente la ganancia del sensor con una resolución muy fina.

Siendo la Electrónica Digital un área de interés actual, continuamente se están desarrollando métodos para el diseño de controladores y para el procesado de imágenes, siendo éste el propósito del proyecto descrito en el presente artículo. Se pueden encontrar trabajos como [4] que describe el desarrollo de un controlador para una cámara CMOS, con aplicación en MAV's (*Micro Air Vehicles*). Lo cual conlleva una serie de requisitos específicos para que el controlador sea útil, las pruebas se realizaron con una tarjeta Altera DE2-115.

Además, existe documentación que proporciona Altera acerca de trabajos realizados en el desarrollo de controladores de dispositivos externos a la FPGA. En [5] se presenta un método para implementar un controlador de un sensor de imagen y un sistema capaz de procesar video en alta definición.

En muchas de las ocasiones se requiere implementar en hardware un controlador que ofrezca la posibilidad de seleccionar la longitud de onda a procesar y a la vez sea lo suficientemente flexible para ser adaptado a especificaciones de otras aplicaciones. En este trabajo se describe el diseño de un controlador que cumple con estas características. En la siguiente sección se describe brevemente el sensor empleado, además de los requerimientos en hardware necesarios, específicamente lo que respecta a la capacidad de almacenamiento. En la tercera sección se presentan las características generales de la tarjeta de desarrollo de Altera empleada. La descripción del sistema implementado se presenta en la cuarta sección. Los resultados obtenidos son mostrados en quinta sección. Finalmente, se presentan las conclusiones y potenciales aplicaciones a futuro del sistema desarrollado.

2. Cámara CCD

El arreglo de pixeles del sensor CCD TRDB- D5M (Fig. 1) consiste en una matriz de 2752 columnas por 2004 filas, direccionadas por columna y fila. La dirección (columna 0, fila 0) representa la esquina superior derecha del arreglo, como se muestra en la Fig. 2.



Fig. 1. Sensor de imagen TRDB D5M.

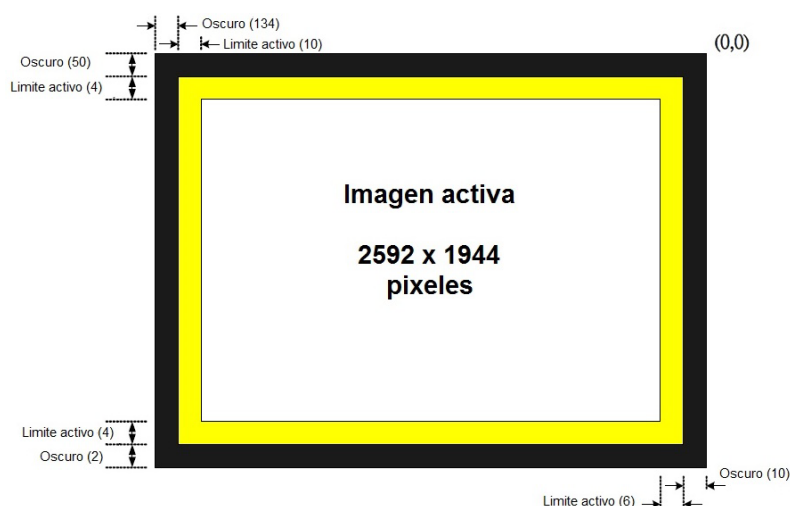


Fig. 2. Descripción del arreglo de píxeles.

El arreglo consiste en una región activa de 2592 columnas por 1944 filas representando la imagen de salida por defecto. Dicha zona activa está rodeada por una región también activa (representada en amarillo en Fig. 2). A la vez se tiene una región límite que representa un marco de píxeles negros. La región límite puede ser usada para prevenir efectos de borde cuando se realiza algún procesamiento y se obtenga como resultado una imagen de 2592 x 1944.

Los píxeles a la salida de la cámara están ordenados en un formato de patrón Bayer, el cual consiste en cuatro colores: verde1, verde2, rojo y azul (G1, G2, R, B). Por default se tiene la primera fila de salida alterna entre G1 y R, y la segunda fila alterna entre los píxeles B y G2. El verde1 y verde2 tienen el mismo filtro de color (Fig. 3).

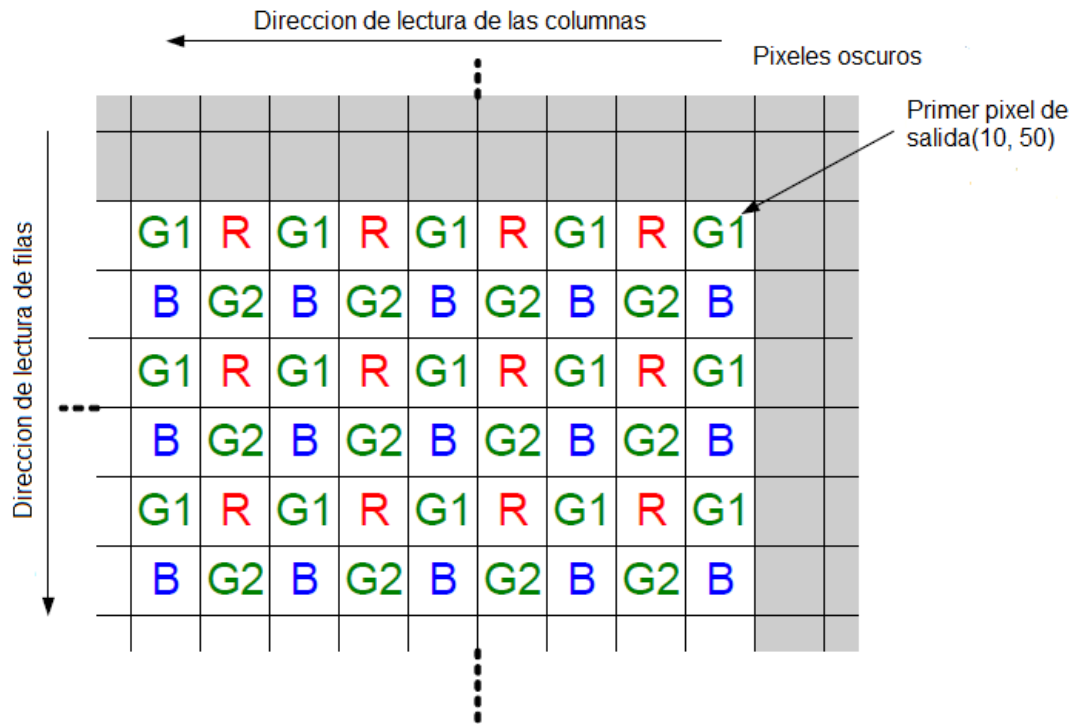


Fig. 3. Detalle del patrón de píxeles (esquina superior derecha).

Las imágenes de salida se dividen en cuadros, que a la vez se dividen en líneas. Por default el sensor produce 1944 filas por 2592 columnas que equivalen a cuadros por líneas. Las señales *FRAME_VALID* y *LINE_VALID* indican el límite entre cuadros y líneas, respectivamente.

La señal *PIXCLK* puede ser usada como un reloj que indica cuando el dato de salida de la cámara cambia. Por cada ciclo de *PIXCLK* el sensor arroja al puerto de salida *D* un dato de 12 bits que equivale a un pixel. Cuando ambas señales *FRAME_VALID* y *LINE_VALID* coinciden en nivel lógico alto entonces el pixel es válido. Los ciclos de la señal *PIXCLK* que ocurren cuando *FRAME_VALID* está en nivel lógico bajo son llamados “vacío vertical”. Los ciclos de *PIXCLK* que ocurren solo cuando *LINE_VALID* está en cero son llamados “vacío horizontal” (Fig. 4).

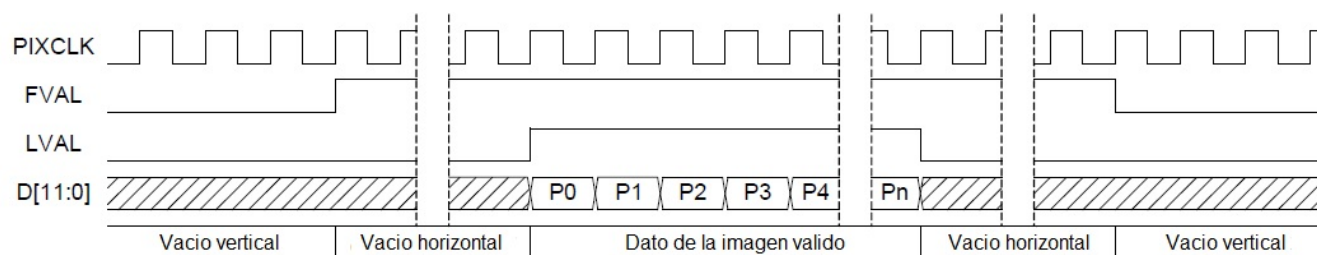


Fig. 4. Diagrama de tiempo de la salida por defecto.

Los parámetros descritos son válidos cuando los registros internos de la cámara están en su valor por defecto. La lectura y escritura de los registros se realizan en la TRDB D5M a través de dos líneas de comunicación serial. La TRDB D5M es un esclavo en la interfaz serial y es controlado por un reloj serial (*SCLK*), que es generado por un maestro. El dato de salida se transfiere vía serial a través del dato serie (*SDATA*). La comunicación entre la cámara CCD y la lógica de control del FPGA se lleva a cabo mediante el protocolo de comunicación I2C [6].

La TRDB D5M cuenta con 256 registros, de los cuales solo algunos son modificables ya que la mayoría de estos son un medio de almacenamiento para la lógica de la propia cámara y no pueden ser modificados. Dentro de los registros a los cuales se tiene acceso se puede programar: el ancho y largo de la imagen, establecer la ganancia de colores, realizar inversión de las señales de control, configurar el método de captura, decimar los datos de salida y aumentar o disminuir internamente la frecuencia de trabajo de la cámara.

Una de las acciones de control sobre la cámara establecidas en el proyecto es reducir el tamaño de la imagen de salida. Esta decisión se tomó con la finalidad de reducir la complejidad de los potenciales algoritmos de procesamiento a emplear. Además se consideró una frecuencia de operación del controlador lo suficientemente alta para una lograr una adecuada velocidad de transferencia de datos para su exhibición en un monitor VGA.

3. Tarjeta Altera DE2-70

El sistema empleado para la implementación del controlador de la cámara fue la tarjeta DE2-70 de Altera (Fig. 5). Esta tarjeta provee al diseñador un conjunto de herramientas dentro de un ambiente de desarrollo completo y de alta calidad, el cual también incluye: un software de programación, diseños de referencia, cables, y sobre todo el hardware reconfigurable.

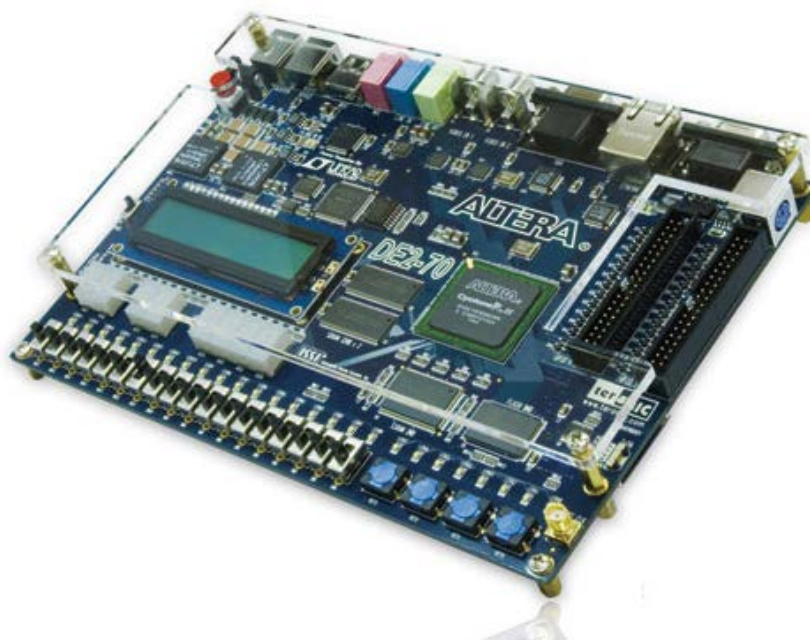


Fig. 5. Altera DE2-70.

La tarjeta Altera DE2-70 está equipada con un dispositivo FPGA Cyclone II 2C70 que cuenta con el hardware necesario para aplicaciones multimedia, almacenamiento de datos, e interfaces de red. La tarjeta ofrece un gran conjunto de características que la hacen ideal para ser usada en laboratorios de universidades y para prácticas didácticas, para una variedad de proyectos de diseño, así como para el desarrollo de sofisticados sistemas digitales. Además Altera provee material de apoyo para la tarjeta DE2-70, incluyendo tutoriales, ejercicios de laboratorio y demostraciones ilustrativas [7].

4. Sistema propuesto

El controlador propuesto se implementó mediante las herramientas de Altera, las cuales también fueron útiles para demostrar el correcto funcionamiento del mismo. Como salida del controlador se tiene la imagen exhibida en un monitor VGA. Todo el sistema desarrollado, a excepción de un componente, fue descrito en lenguaje VHDL [8-9].

El controlador implementado genera las señales necesarias para configurar cada uno de los registros que son necesarios para alguna aplicación en específico. La primera acción que se realiza es configurar la cámara para una reducción de imagen de salida. Dicha configuración se realizó para lograr una imagen de 400 filas por 400 columnas. El tamaño de la imagen se determinó en base a los recursos de memoria interna con que se cuenta en el FPGA; ya que a una mayor resolución los tres mapas (RGB) exceden la capacidad de almacenamiento interno del FPGA. Además del establecimiento de la resolución de salida, se aumentó la ganancia de cada uno de las longitudes de onda y se configuró la cámara en modo “*snapshot*”. En dicho modo la captura se controla con una señal activa en bajo de la cámara (TRIGGER). El resto de los registros se conservaron en su valor por defecto, para lo anterior se describió un componente en lenguaje Verilog. Como resultado se tiene un bloque cuyo funcionamiento es independiente del resto del sistema y que es controlado solamente por el reloj principal de la tarjeta utilizada, a este bloque se le nombró CCD_Config (Fig. 6).

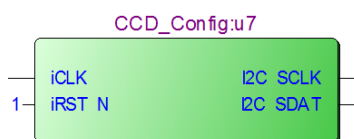


Fig. 6. Bloque para configuración de cámara CCD.

Para comprobar la función de configuración del tamaño de la imagen de salida de la cámara, se describió un bloque que, con las señales *FRAME_VALID*, *LINE_VALID* y *PIXCLK*, calcula el número de filas y de columnas de la imagen de salida de la cámara, el resultado se muestra en los LEDs disponibles en la tarjeta (Fig. 7).

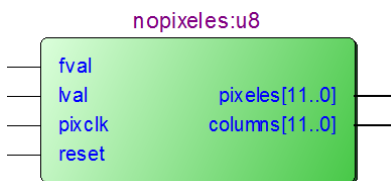


Fig. 7. Bloque para comprobar la configuración de la cámara CCD.

Una vez que la cámara está configurada se procede con la captura de las imágenes. Para el almacenamiento de una longitud de onda en específico se utilizó una memoria de doble puerto con una capacidad de 40000 localidades (200 pixeles x 200 pixeles) de 8 bits. El resultado de salida de la cámara, que es de 12 bits, se trunca a 8 para lograr almacenar una mayor cantidad de pixeles. Aunque la cámara esté configurada para capturar una imagen de 400 filas por 400 columnas, al almacenar solo la información de una longitud de onda el resultado será una imagen almacenada a la mitad de filas y la mitad de columnas.

El almacenamiento de la imagen capturada se realiza por un puerto de una memoria de doble puerto, donde la señal *we* (writeenable) controla la longitud de onda que es almacenada. Dicha señal, a la vez, se selecciona mediante dos pines de entrada (*mapselect*) del bloque que contiene la memoria (Fig. 8). El dato de salida de la memoria lo determina la dirección del segundo puerto de la memoria, esta dirección la controla otro bloque que es el encargado de generar las señales para desplegar el resultado en un monitor VGA.

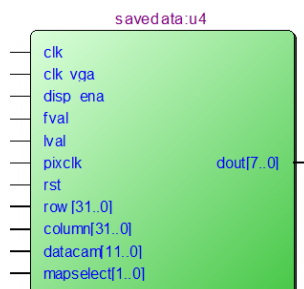


Fig. 8. Bloque encargado del almacenar la longitud de onda deseada.

Para que exista sincronismo, entre el monitor VGA y el dato que está almacenado en la memoria, se implementó un controlador (Fig. 9) que genera un contador de filas y otro de columnas de la imagen a desplegar. Dichas contadores determinan la dirección a leer de la memoria.

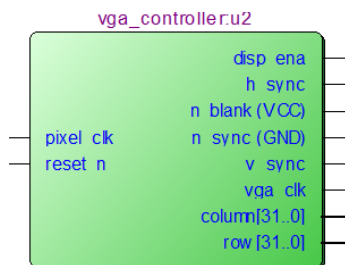


Fig. 9. Controlador para monitor VGA.

De igual manera, es importante un bloque que genere las señales de reloj apropiadas para controlar el monitor VGA. Se utilizó un divisor de frecuencia para reducir la frecuencia de reloj disponible en tarjeta, esto es reducir de 50 MHz a 25 MHz (Fig. 10).

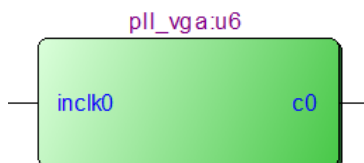


Fig. 10. Bloque pll (phase locked loop) para dividir la frecuencia.

El sistema completo funciona de tal manera que mientras la señal *TRIGGER* se active se guardará la imagen. La dirección de escritura del dato y la señal *we* son entradas al primer puerto de la memoria RAM, y al mismo tiempo se asigna una dirección donde leer. De acuerdo a lo anterior por un puerto se escribe y por otro se lee el dato. En el sistema (Fig. 11) se utilizan

tres bloques *savedata* (Fig. 7), de tal forma que cada uno se configura para capturar una longitud de onda diferente, de esta manera el usuario puede seleccionar la longitud de onda deseada a desplegar en el monitor o incluso desplegar una imagen a color.

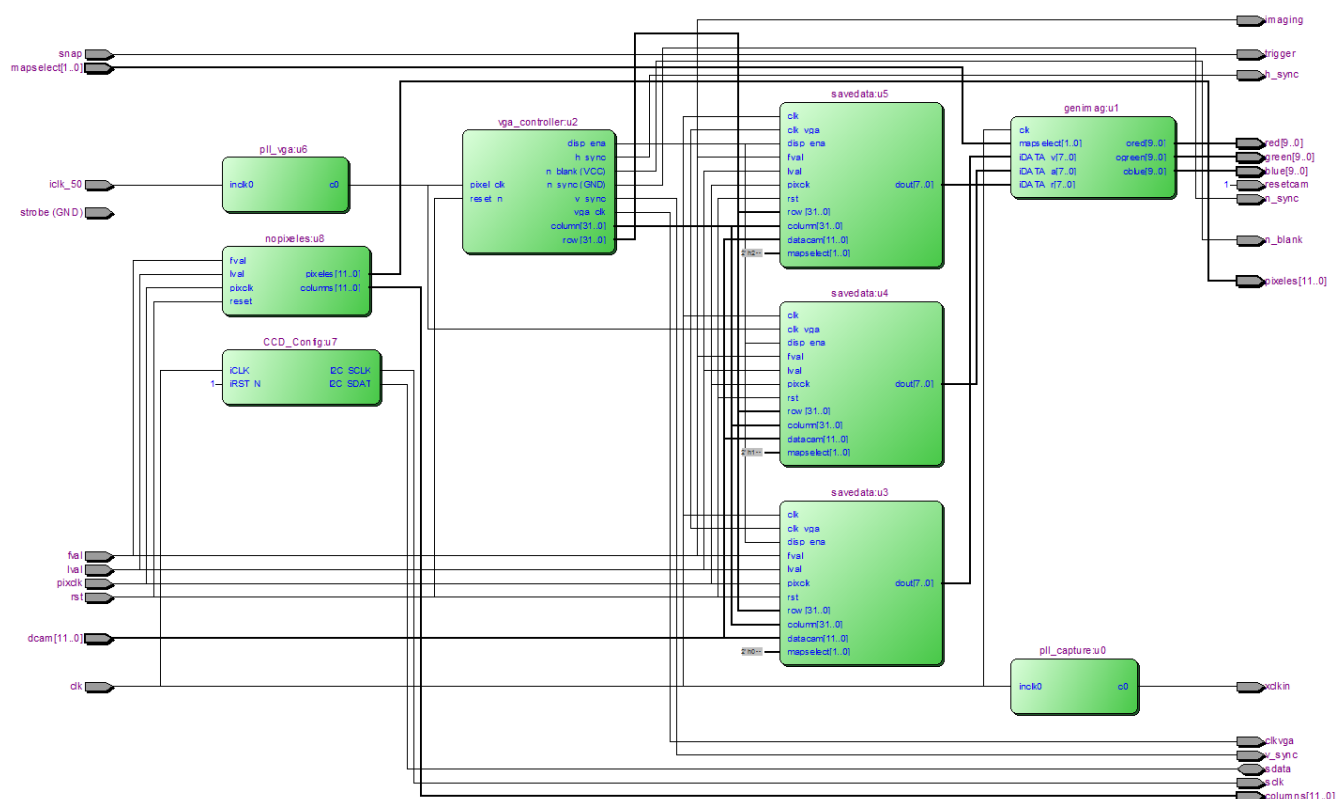


Fig. 11. Sistema implementado en FPGA.

5. Resultados

En la Fig. 12 se ilustra el resultado de simulación del bloque encargado de la configuración de la cámara. En la simulación se muestra la transmisión mediante el protocolo I2C de un dato de 8 bits (01000101) a la dirección arbitraria 0x059. En la misma figura se indican las señales *SCLK* y *SDATA* de TRDB D5M, en donde las señales en color azul representan líneas de alta impedancia. También se incluye en simulación un contador de bits transmitidos.

El proceso para almacenar el dato en la memoria interna dentro del FPGA se ilustra en la Fig. 13, donde a partir de las señales del sensor (*LVAL* y *FVAL*) se generan tres señales *writable*, cada una para diferente mapa de color. La señal de entrada a la memoria *we* es establecida por el usuario mediante dos interruptores disponibles en la tarjeta. Dentro de las señales más relevantes se tiene *datacam* que representa el dato emitido por la cámara, *xaddress* que contiene la dirección de la memoria RAM donde se almacena el dato y las señales *we_rojo*, *we_verde*, *we_azul* que determinan el momento cuando el dato de la cámara debe ser almacenado.

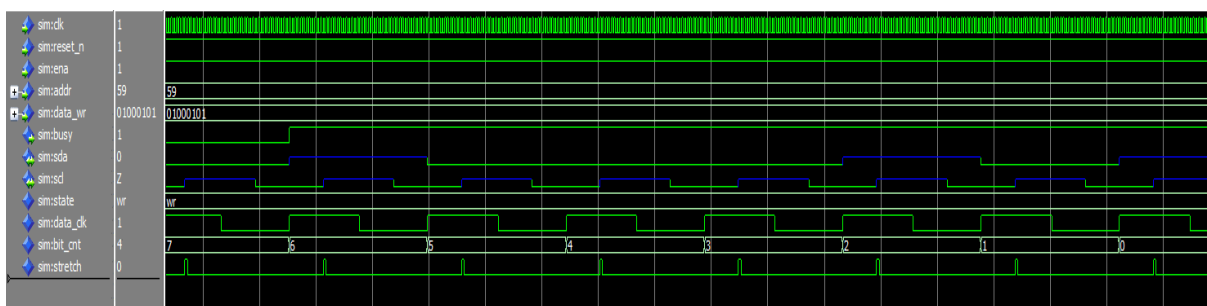


Fig. 12. Simulación para la configuración de la cámara.

El línea en amarillo en Fig. 13 indica el momento cuando el primer dato de la cámara (0x027), una vez que las señales *LVAL* y *FVAL* están en alto, coincide con la señal *we* del mapa rojo y el flanco de subida del reloj. Una vez que se cumplen estas condiciones el dato es almacenado en la dirección indicada. La dirección de escritura se incrementa en uno en cada flanco positivo de la señal *we*.

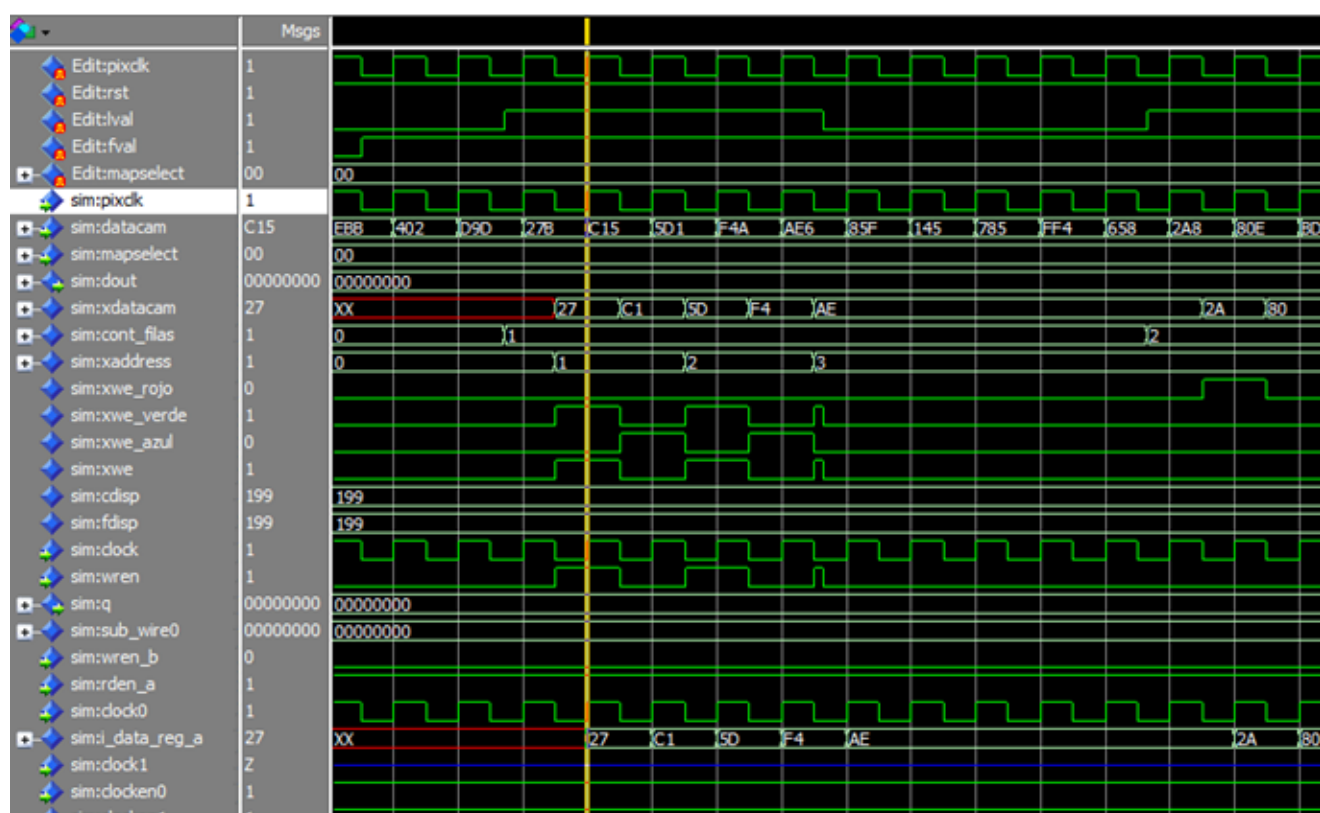


Fig. 13. Simulación para la escritura de la imagen en la memoria RAM.

En la Fig. 14 se muestra el correcto funcionamiento del controlador a través de una imagen capturada con la cámara CCD a una resolución de 200 x 200 píxeles. En este caso la imagen se muestra mediante los tres mapas de color. De igual manera en las Fig. 15, 16 y 17 se presentan la misma imagen con la selección de los mapas de colores verde azul y rojo, respectivamente.



Fig. 14. Imagen original desplegada a color.



Fig. 15. Imagen capturada con longitud de onda verde.



Fig. 16. Imagen capturada con longitud de onda azul.



Fig. 17. Imagen capturada con longitud de onda rojo.

Con el fin de comprobar visualmente la correcta captura de los mapas de colores, se modificó la descripción del sistema de tal manera que el color seleccionado se despliegue simultáneamente en los tres colores disponibles en el monitor VGA. Con lo anterior se obtiene una imagen en escala de grises resaltando las zonas con el color seleccionado. En las Fig. 18,

19 y 20 se presentan las imágenes obtenidas con el sistema modificado para los colores verde, azul y rojo, respectivamente.



Fig. 18. Imagen capturada con longitud de onda verde en tres mapas.



Fig. 19. Imagen capturada con longitud de onda azul en tres mapas.



Fig. 20. Imagen capturada con longitud de onda rojo en tres mapas.

6. Conclusiones

El punto de partida en el procesamiento digital de imágenes es el estudio de las técnicas o algoritmos propios para manipular las imágenes, cuya implementación se realiza en herramientas de software, como MATLAB; donde la captura de imágenes se realiza mediante cámaras cuyo controlador es completamente del tipo software. Por otro lado el implementar una aplicación de procesamiento digital de imágenes a nivel hardware requiere como primera etapa el desarrollo del controlador apropiado para la cámara utilizada.

De acuerdo a los resultados obtenidos y presentados en este artículo, se concluye que la propuesta de controlador, diseñado e implementado en hardware reconfigurable, cumple funcionalmente con las especificaciones planteadas al inicio del proyecto. La culminación exitosa del presente proyecto dio como resultado un controlador implementado en hardware reconfigurable que permite la captura de imágenes de la cámara CCD utilizada.

Cabe resaltar que previo al desarrollo de este trabajo se tenía disponible un controlador de la cámara CCD, el cual forma parte de la tarjeta de desarrollo empleada. Sin embargo, dicho controlador es muy específico y escasamente documentado. Con lo anterior, cuando surge la necesidad de hacer uso de la cámara en alguna aplicación propia es un tanto difícil adaptar mediante modificaciones el diseño original. Por ejemplo la aplicación descrita de capturar y

exhibir los mapas de colores de la imagen. El desarrollo propio del controlador de la cámara, propuesto en este artículo, permitirá hacer su uso en un rango de aplicaciones más abierto en el área de procesamiento de imágenes. En particular el controlador desarrollado se empleará como punto de partida para el procesamiento de imágenes con aplicación en reconocimiento de patrones.

7. Referencias

- [1] R. Srinivasan, K. Anupama, Suneeta, S. K. Saha, and Aditya Rao, "FPGA Based ASM implementation for CCD Camera Controller". *International Conference on Emerging Trends in Electronic and Photonic Devices & Systems*, 2009.
- [2] A. B. Benhouzid, A. Rachedi, and K. Laidi, "A New Micro-Satellite CCD Camera Controller Design". *Recent Advances in Space Technologies*, 2005.
- [3] Tae-Hwan Oh., and Seung-Hoon Lee, "Single-Chip CMOS CCD Camera Interface Based on Digitally Controlled Capacitor-Segment Combination". *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, 47(11), Nov. 2000.
- [4] O. Ranganathan and Abdul Imran Rasheed, "Desing and Development of Camera Interface Controller With Video Preprocessing Modules on FPGA for MAVs". *SASTECH Journal*, vol. 12, no 1, April 2013.
- [5] "Building an IP Surveillance Camera System with a Low-Cost FPGA". Altera. 2012.
- [6] TRDB-D5M Hardware specification. Terasic. 2009.
- [7] Altera DE2-70 User's Manual. Altera. 2009.
- [8] K. C. Chang, Digital design and modeling with VHDL and synthesis. *IEEE Computer Society Press*, Los Alamitos California, USB, ISBN 0-8186-7716-3, 1997.

- [9] J. R. Armstrong, F. G. Gray, "VHDL design representation and synthesis". 2nd ed., *Prentice Hall Modern Semiconductor Design Series*, New Jersey, USA, ISBN 0-13-021670-4, 2000.

8. Autores

Ing. Juan Moisés Arredondo Velázquez obtuvo su título de Ingeniero en Mecatrónica de la Universidad Tecnológica del Norte de Guanajuato, y actualmente está cursando la Maestría en Ciencias en Electrónica en el Instituto Tecnológico de Celaya.

Dr. José Javier Díaz Carmona, obtuvo su título de Ingeniero en Electrónica en el Instituto Tecnológico de Celaya, posteriormente los títulos de Maestro y Doctor en Ciencias en Electrónica en el Instituto Nacional de Astrofísica Óptica y Electrónica, actualmente se desempeña como docente e investigador en el departamento de Ing. Electrónica del Instituto Tecnológico de Celaya.

Dr. José Alfredo Padilla Medina, obtuvo su título de Ingeniero en Electrónica en el Instituto Tecnológico de Celaya, posteriormente su título de Maestro en Ciencias en Electrónica en la Universidad de Guanajuato, y su título de Doctor en Ciencias en Óptica en el Centro de Investigación en Óptica en León Gto, actualmente se desempeña como docente e investigador en el departamento de Ing. Electrónica del Instituto Tecnológico de Celaya.

Dr. Agustín Ramírez Agundis, obtuvo sus títulos de Ingeniero en Eléctrica y Maestro en Ciencias en la Universidad de Guanajuato, y su título de Doctor en Ciencias en Electrónica en la Universidad Politécnica de Valencia España, actualmente se desempeña como docente e investigador en el departamento de Ing. Electrónica del Instituto Tecnológico de Celaya.

Dr. Noé Vela Aguirre obtuvo sus títulos de Ingeniero en Eléctrica y Maestro en Ciencias en la Universidad de Guanajuato, y su título de Doctor en Ciencias en Electrónica en la Universidad Politécnica de Valencia España, actualmente se desempeña como docente e investigador en el departamento de Ing. Electrónica del Instituto Tecnológico de Celaya.